

## Victorian 6502 User Group Newsletter

# KAOS

For People Who Have Got Smart

HARDWARE .. .. . DAVID ANEAR  
SOFTWARE .. .. . JEFF RAE  
FORTH .. .. . DAVID WILSON  
AMATEUR RADIO .. .. . ROD DRYSDALE VK3BYU  
EDUCATION .. .. . JEFF KERRY  
LIBRARY .. .. . RON KERRY  
TAPE LIBRARY .. .. . JOHN WHITEHEAD  
DISK LIBRARY .. .. . WARREN SCHAECH (B.H.)  
NEWSLETTER .. .. . IAN EYLES  
SYM. .. .. . BRIAN CAMPBELL  
SECRETARY .. .. . ROSEMARY EYLES

---

OSI	SYM	KIM	AIM	UK101	RABBLE 65
-----	-----	-----	-----	-------	-----------

---

Registered by Australia Post  
Publication No. VBG4212

ADDRESS ALL CORRESPONDENCE TO 10 FORBES ST., ESSENDON, VIC. 3040

Vol.4 No.2

November 1983

Well it is that time of the year again. You will find your new membership form enclosed with this issue of the newsletter. The membership fee will be the same, \$15 for Australian members, \$20 for New Zealand and \$25 for overseas members. If you have not paid your fees by the end of January 1984, you will not receive your February newsletter.

As usual we will not be having a December meeting, but our November meeting will be a family day, there will be a BBQ in the school grounds starting at 11am. We are hiring a couple of large BBQs and supplying bread and tomato sauce, you bring your family, food and drink (alcohol in moderation please). There is cover if it is wet and there is plenty of play equipment to keep the children amused. For addicts who can't stay away from their computers, the downstairs hall will be open and there will be a short meeting at about 2pm.

The fate of OSI still seems to be up in the air. We received a newspaper clipping from Mike Keryan in America stating that the Aurora division of OSI had closed and the Bank of America had taken over all the stock. Then we received the latest PEEK 65 and the editor, Al Peabody, listed the rumours he has heard, OSI is looking for new premises, OSI is being taken over by a Swedish company ..... Your guess is as good as ours. One thing we do know, we are still getting new members and according to Norm Bate there are a lot of Ohio owners around who haven't heard of KAOS.

The next meeting will be held on Sunday 27th November at 2pm at the Essendon Primary School which is on the corner of Raleigh and Nicholson Streets, Essendon.

The closing date for items for the December newsletter will be 2nd December.

### INDEX

A.L.F.O. ....	7
Assembler - ROMable .....	10
Competition .....	8
EXMON Plus .....	2
For Sale .....	4
Machine Language pt 17 .....	9
Meeting - KAOS .....	4
Meeting - NSW .....	10
Memory Faults .....	5

My Superboard pt 8 .....	2
NMhz Fix .....	13
Notes from the West .....	11
Printer Board Fix .....	12
Screen Dump - 3.3 .....	15
SUPERBOARD .....	5
T.A.B. ....	12
Token LOAD/SAVE .....	8
Treasure Chest .....	6

MY SUPERBOARD series 2. Part 8.

*by John Whitehead*

EXMON PLUS -- Continuing on from part 6, I have another 2k EPROM at \$E000 to \$E7FF which contains a collection of machine code routines. These are used the same way as EXMON and many use EXMON subroutines. Because of this I have changed my EXMON 'J' key to goto \$E000 so that EXMON is initialised before entering EXMON PLUS.

Below is a list of the routines in alfa order and for the shorter ones I will be giving an assembly listing. I can't give the whole listing as it takes up 24 A4 pages, but EXMON PLUS is available in a 2716 EPROM for \$10 or EXMON and EXMON PLUS in a 2732 from \$E000 to \$EFFF for \$12.

A = set real time clock Alarm.  
C = Cold start BASIC using any block of ram specified.  
ctrl B = screen format swap.  
E = goto Exmon.  
K = start real time Klock.  
M = Monitor.  
N = hex search that allows don't care (wild card) hex numbers.  
O = goto paged rOm at \$8000.  
P = comPare two blocks of memory and print the differences.  
Q = disassemble from and to specified addresses - for use with a printer.  
R = Relocate a BASIC program from one place in memory to another, ready to run it there.  
S = simple, non destructive, memory test.  
T = Teletype routines.  
U = Un-crash BASIC after it's crashed.  
W = Warm start a BASIC program at a specified address.  
Y = micro processor axY trace as KAOS July 83.  
Z = display 8 bytZ of memory in hex and ASCII

One use for C, W and R is to be able to have two different copies of the same BASIC program in memory at the same time for comparison while making changes.

As I call some of the non EXMON dependent routines from other programs, eg Teletype from Word processor, I had to find a way to keep the starting addresses of these in the same place, even if I reassemble EXMON PLUS. This is to save me having to re-burn all my EPROMs every time I update one of them. I did this by reserving the first 48 bytes for a Jump table. eg:-

E000 4C30E0 JMP MENU	The order in the table
E003 4C3DE1 JMP XMONU	is never changed.
E006 4C4DE4 JMP IPAGE	Only the JUMP TO address.

Below are the EXMON subroutine entry point differences between my version and the original, that will need to be changed in the assembly listings if you are using the original EXMON.

Addresses \$0800 to \$0AFE (\$E800 to \$EAFE) are unchanged.

EB19 = 0B07	EB23 = 0B11	EB2E = 0B1C
EB48 = 0B36	EB4A = 0B38	EB5A = 0B48
EC63 = JSR 0AAC and JSR 0C51		EC66 = 0C51

4520 E385		; Compare memory blocks (P)	5240 E419		; Screen swap (ctrl B)
4530 E385 205AEB	COMP	JSR \$EB5A ;get addr	5250 E419 AD1A02	CTRLB	LDA \$021A
4540 E388 2019EB		JSR \$EB19 ;LF,CR	5260 E41C C900		CMP #\$00
4550 E38B B1DC	P1	LDA (\$DC),Y	5270 E41E F012		BEQ B1
4560 E38D D1DA		CMP (\$DA),Y	5280 E420 A200		LDX #\$00
4570 E38F F027		BEQ P2	5290 E422 8E1A02		STX \$021A
4580 E391 2066EC		JSR \$EC66 ;space	5300 E425 E8		INX
4590 E394 A5DB		LDA \$DB	5310 E426 86FB		STX \$FB
4600 E396 20ACEA		JSR \$EAAC ;lbyte to 2 char	5320 E428 A9FB		LDA #\$FB
4610 E399 A5DA		LDA \$DA	5330 E42A 8D1B02		STA \$021B
4620 E39B 20C3E3		JSR P3	5340 E42D 2000FB		JSR \$FB00
4630 E39E B1DA		LDA (\$DA),Y	5350 E430 D00F		BNE B2
4640 E3A0 2063EC		JSR \$EC63	5360 E432 A969	B1	LDA #\$69
4650 E3A3 2066EC		JSR \$EC66	5370 E434 8D1A02		STA \$021A
4660 E3A6 A5DD		LDA \$DD	5380 E437 A9FF		LDA #\$FF
4670 E3A8 20ACEA		JSR \$EAAC	5390 E439 8D1B02		STA \$021B
4680 E3AB A5DC		LDA \$DC	5400 E43C A900		LDA #\$00
4690 E3AD 20C3E3		JSR P3	5410 E43E 8D00D8		STA \$D800
4700 E3B0 B1DC		LDA (\$DC),Y	5420 E441 4C30E0	B2	JMP MENU
4710 E3B2 20ACEA		JSR \$EAAC ;byte to 2 char			
4720 E3B5 2019EB		JSR \$EB19 ;LF,CR			
4730 E3B8 2048EB	P2	JSR \$EB48 ;inc double add&comp	8980 E712		; show 8 bytes in hex and ASCII (Z)
4740 E3BB E6DA		INC \$DA	8990 E712 2023EB	ZPLUS	JSR \$EB23
4750 E3BD D0CC		BNE P1	9000 E715 2019EB		JSR \$EB19
4760 E3BF E6DB		INC \$DB	9010 E718 A5DB	Z1	LDA \$DB
4770 E3C1 90C8		BCC P1	9020 E71A 85DD		STA \$DD
4780 E3C3 20ACEA	P3	JSR \$EAAC ;byte to 2 char	9030 E71C 20ACEA		JSR \$EAAC
4790 E3C6 A93D		LDA #\$3D	9040 E71F A5DA		LDA \$DA
4800 E3C8 4CEEFF		JMP \$FFEE	9050 E721 85DC		STA \$DC
			9060 E723 A8		TAY
			9070 E724 20ACEA		JSR \$EAAC
			9080 E727 2066EC		JSR \$EC66
			9090 E72A 2066EC		JSR \$EC66
			9100 E72D A208		LDX #\$08
			9110 E72F A000		LDY #\$00
			9120 E731 B1DA	Z2	LDA (\$DA),Y
			9130 E733 2063EC		JSR \$EC63
			9140 E736 E6DA		INC \$DA
			9150 E738 D002		BNE Z3
			9160 E73A E6DB		INC \$DB
			9170 E73C CA	Z3	DEX
			9180 E73D D0F2		BNE Z2
			9190 E73F 2066EC		JSR \$EC66
			9200 E742 A208		LDX #8
			9210 E744 A000		LDY #0
			9220 E746 B1DC	Z4	LDA (\$DC),Y
			9230 E748 C910		CMP #\$10
			9240 E74A 9004		BCC Z5
			9250 E74C C97F		CMP #\$7F
			9260 E74E 9002		BCC Z6
			9270 E750 A95B	Z5	LDA #'
			9280 E752 2061E8	Z6	JSR \$E861
			9290 E755 E6DC		INC \$DC
			9300 E757 D002		BNE Z7
			9310 E759 E6DD		INC \$DD
			9320 E75B CA	Z7	DEX
			9330 E75C D0E8		BNE Z4
			9340 E75E 2019EB		JSR \$EB19
			9350 E761 2053E8		JSR \$E853 ;get key
			9360 E764 C90A		CMP #\$0A
			9370 E766 F0B0		BEQ Z1
			9380 E768 60		RTS

4820 E3CB		; Disassemble FROM-TO (Q)
4830 E3CB 202EEB	DISFT	JSR \$EB2E ;get double addr
4840 E3CE A5DC		LDA \$DC
4850 E3D0 8555		STA \$55
4860 E3D2 A5DD		LDA \$DD
4870 E3D4 8556		STA \$56
4880 E3D6 2019EB		JSR \$EB19 ;LF,CR
4890 E3D9 85D8		STA \$D8
4900 E3DB 209CE9	Q1	JSR \$E99C
4910 E3DE 20A3E9		JSR \$E9A3
4920 E3E1 A556		LDA \$56
4930 E3E3 85DD		STA \$DD
4940 E3E5 A555		LDA \$55
4950 E3E7 85DC		STA \$DC
4960 E3E9 204AEB		JSR \$EB4A ;inc double addr
4970 E EC 4CDBE3		JMP Q1

GEORGE WILL NOT BE BEATEN!

If you want a good low cost dot matrix printer, you can now get the Alpha 80, JP-80, CP-80 or whatever-80 at the incredibly low price of \$399 inc. tax!

WHILE STOCKS LAST

Delivery to almost anywhere in Australia for \$10.00  
.....

MITSUBISHI 8" SLIMLINE DISK DRIVES

M2896-63 Double sided/Double density for only \$540 inc. tax!  
Only 4 in stock.

COMP-SOFT 235 Swan St, Richmond 3121 Ph 03 426 5269 03 429 9686

THE MEETING WAS KAOS  
*by King Corky*

This month's meeting was opened, unceremoniously, by our new chairman, Warren Scaeché. We don't stand for too much pomp and ceremony at our meetings, all they do is slow things down and achieve no real purpose other than to boost the ego of those requiring it. Anyway the most important news is the BBQ to be held at the school on 27th November, our normal meeting day, starting time is 11am. We are hiring some big barbecues and providing bread and tomato sauce, BYO everything else. Liquid refreshment may be brought along but please keep to a minimum as this is a family outing.

The news from Queensland, the land of Jo, ex-AIM keyboards for \$30. Changes are required to suit Rabble. Paul Dodd announced that there will be some new and exciting mods to COMP-SOFT, a CPM board soon to be available for the Rabble, a new DOS and new BASIC and they are also working on a parallel transfer bus, (ask Paul about that one).

Ray Gardiner demo'd the Rabble video driver with numerous screen windows, 64/80 column, a buffered line input routine, courtesy of Paul and Michael, and SDP, a Software Development Package, which is available from George for the Rabble.

Ron Kerry has dropped his endeavours to import the 1st to 5th books of OSI. The distributors won't come to the party with rates low enough to make bulk buying worthwhile. John Whitehead demo'd his logic simulator routine which displays a timing circuit type of graph output for a given set of logic gate input states. I think you can even work backwards to give a logic setup for a required end result. David Anear showed some mini-floppy head cleaning disks that will last as long as the bottle of fluid they come with. At 2 drops per clean, this is quite some time. Cost, approximately \$10. He also announced a new Mostek dynamic memory chip that has a built-in dilithium battery that should hold memory contents for about a year.

Finally, David Dodd has a new Assembler, for more details see David's article on Machine Language. See you at the BBQ. Bye for now.

---

FOR SALE

C4P, with 32K, 2MHz, one 8" D/S D/D floppy disk drive, B&W TV modified for direct video, 2 joysticks, much documentation. Owner moving to N.S.W., so please buy it! A bargain at a negligible \$700 O.N.O.  
Contact Peter McLennan

Superboard in case with Rabble Ozi expansion board, 40K memory, Dabug 3, BASIC 4, lots of documentation and programs \$300 O.N.O.  
Contact Kev Goffey

Power supply 5V, 10amp regulated in case. Transformer has 2 X 14V-1A windings for F.D.C.  
\$50.00 O.N.O. Contact T. Foulds

---

ATTENTION ALL VIC-20 OWNERS

Now at last there is a reasonable priced method of expanding your Vic-20 computer. This expansion board provides 8K of RAM plus an excellent machine language monitor. It is fully self contained. NO Motherboard system is needed. The PCB is double sided with plated through holes, and gold flashed edge connector. Full address and data line buffering is incorporated.

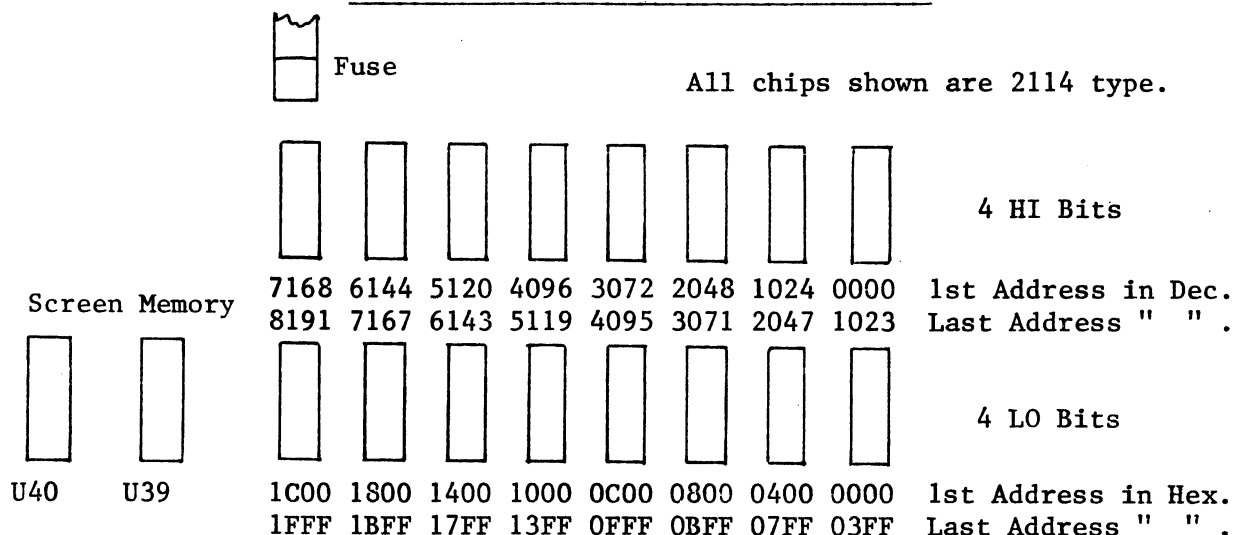
Lowmon-20 plus 8K RAM Expansion board comes completely assembled and tested with 90 days conditional warranty.  
Price including post and packing in Australia is \$130.  
Write for more details to:- LOW-SOFT, P.O. Box 340, St Agnes, SA 5097.

# Superboard

November, 1983.

Newsletter of the Ohio Superboard User Group, 146 York Street, Nundah, 4012.

## SERVICING MEMORY FAULTS ON THE O.S.I.



OK, so you think a memory chip is at fault. Your computer says 2303 bytes free on a coldstart. The above diagram should help to locate the faulty chip. First, it is important to know that the figure 2303 is the last location which passed the OSI's memory check. The first 768 bytes is the memory used by Basic, and is not part of the test, so to find the pair of memory chips where the test failed, add  $2303 + 768 + 1 = 3072$ . This tells us that the fault may be in one or both of the chips in the 4th row from the right of the diagram. Which one? To find out we need to know whether the HI bits or LO bits caused the failure. To test the LO bits, try this program:-

```
10 FOR R=0 TO 15:POKE 3072,R
20 IF PEEK(3072)<>R THEN PRINT PEEK(3072)
30 NEXT
```

If no numbers are printed, or the numbers printed are all above 15, then the HI bits chip is the probable fault. Otherwise try the LO bits chip. If replacement, or swapping with a chip from the 7168 to 8191 rows does not fix the problem, then check the underside of the board for bad solder joints. Usually bad joints or a crook socket will give intermittent memory check errors or different bytes free on different tests of the coldstart routine. If all seems OK under the board, then replace the 74LS138 decoder, U22.

Video memory (screen memory) faults usually give regular patterns on the screen after a screen clear, and incorrect characters displaying when you type. Try swapping U39 and/or U40 with the chips in the 7168 to 8191 row.

Should zero page memory be faulty, your computer will hang up during a coldstart and may not even work in the machine code mode. Again the solution is to swap the chips in the 0000 to 1023 row with the 7168 to 8191 row chips.

One of the nasty faults that can occur is when one of the address or data lines are held high or low. Usually, you will get the mixed screen pattern when you switch on, and won't even be able to Break.

# — SUPERBOARD —

Should this occur, you can try one trick before making a trip to the computer doctor. Remove all memory chips except screen memory and the first two ram chips, 0000 to 1023. Try again. No? Remove Basic roms 1,2,3 and 4 and the ACIA. No? Replace screen memory and the two remaining ram chips with four others from your pile. If your computer still won't break in this mode then you need some professional help.

-----

## TREASURE QUEST

What follows is surely one of the shortest Adventure games ever written. The purpose of the article, apart from giving you a fun game to play, is to introduce the concepts of writing such games so that better ones can be written. Next month, we will go through the program, explaining the various routines and variables, and there will be an article on how to set up your own adventure game, and have more control over the events within.

The adventure presented here is a type known as a dungeon. In a dungeon, you have limited control over your character. Some random factors have been written into this one to make it less "tame", and also to shorten it to a length that will not be too great a task to enter from the keyboard. The random factors introduced will have the added effect of making each game different, but mean that very little skill is needed to play. Lady Luck rules!

If your keyboard entry days are over, then John Whitehead has a copy of the program in the KAOS tape library at the normal library rates. OSUG library has a couple for our Library members (with Star Shoot also), at the normal postage costs. People who sent in a solution to our Star Shoot Challenge will have their tapes returned with Treasure Quest on the back.

```
100 PRINT CHR$(26);"TREASURE QUEST":PRINT:PRINT:POKE 11,0:POKE 12,253
110 N=16:M=0:DIM F$(5,5),W$(5),R$(N),C(N,4),D$(4),H(5),A(3),W(5)
120 FOR R=1 TO 5:FOR Y=1 TO 5:READ F$(R,Y):NEXT Y,R
130 FOR R=1 TO 5:READ W$(R):NEXT:FOR R=1 TO N:READ R$(R):NEXT
140 FOR R=1 TO 4:READ D$(R):NEXT:FOR R=1 TO N:FOR Y=1 TO 4:READ C(R,Y):NEXT Y,R
150 FOR R=1 TO 5:W(R)=INT(6*RND(1)+1):H(R)=INT(6*RND(1)+8):NEXT
160 A(1)=0:A(2)=0:T=16:Q=1:H=2:R=INT(5*RND(1)+1):A(3)=R:W(R)=0
170 IF H<0 THEN PRINT:ON Q GOTO 250,260
180 PRINT:PRINT"You find yourself in the";R$(H)
190 IF H=15 THEN PRINT"You drown!":GOTO 450
200 IF H=T THEN PRINT:PRINT"You find the TREASURE !":Q=2:T=0
210 GOSUB 340:IF F=1 GOTO 180
220 GOSUB 460:GOSUB 280
230 R=INT(5*RND(1)+1):IF H(R)<>0 THEN H(R)=INT(13*RND(1)+1)
240 GOTO 170
250 PRINT"You're out without the treasure,go back in!":GOTO 150
260 PRINT"Mission accomplished":IF M<2 THEN PRINT"Great work!":END
270 PRINT M;"men were lost!":END
280 PRINT:PRINT"You can move:-":PRINT
290 FOR R=1 TO 4:IF C(H,R)<>0 THEN PRINT D$(R);",";
300 NEXT R:PRINT:PRINT"Which way ?":X=USR(X)
310 PRINT CHR$(PEEK(531)):PRINT:FOR R=1 TO 4
320 IF CHR$(PEEK(531))=LEFT$(D$(R),1) AND C(H,R)<>0 THEN H=C(H,R):RETURN
330 NEXT R:PRINT:PRINT"YOU CAN'T GO THAT WAY!":GOTO 280
340 FOR R=1 TO 5:IF H(R)=H GOTO 360
350 NEXT R:F=0:RETURN
360 PRINT:PRINT"You see a ";F$(R,1)
370 IF RND(1)>.5 GOTO 420
380 IF Q=2 THEN PRINT:PRINT"You drop the treasure &":Q=1:T=H
```

# — SUPERBOARD —

```

390 PRINT:PRINT F$(R,5)
400 H1=INT(13*RND(1)+1):IF H1=H GOTO 400
410 GOSUB 540:H=H1:F=1:RETURN
420 PRINT F$(R,2):GOSUB 540
430 IF A(1)=R OR A(2)=R OR A(3)=R THEN PRINT F$(R,3):H(R)=0:F=1:RETURN
440 PRINT F$(R,4):PRINT:PRINT"Your FUNERAL can wait!"
450 PRINT:PRINT"Send another man in!":M=M+1:GOSUB 540:GOTO 150
460 PRINT:PRINT"You are carrying:-":PRINT:FOR R=1 TO 3
470 IF A(R)>0 THEN PRINT"a ";W$(A(R))
480 NEXT R:PRINT:FOR R=1 TO 5:IF W(R)=H GOTO 500
490 NEXT R:RETURN
500 PRINT"You find a ";W$(R):PRINT:PRINT"Do you want it ?";
510 X=USR(X):PRINT CHR$(PEEK(531)):IF PEEK(531)=78 THEN RETURN
520 IF A(1)<>0 THEN PRINT"You drop a ";W$(A(1)):W(A(1))=H
530 A(1)=A(2):A(2)=A(3):A(3)=R:W(R)=0:RETURN
540 FOR Y=1 TO 5000:NEXT Y:RETURN
550 DATAGUARD,You fight and,kill him with your SWORD,he kills you!
560 DATAYou both run away!,VENOMOUS SNAKE,it rears and strikes
570 DATAYou destroy it with your STICK,and bites.You're dead!
580 DATAYou run like crazy!,DEADLY SPIDER,It jumps at you
590 DATAYou destroy it with your FLY SPRAY,and stings you.You die!
600 DATAYou retreat with haste,WICKED WITCH
610 DATAShe tries a spell on you,but you use the POTION
620 DATAand turns you into stone,You turn and race away.
630 DATAFIERCE HOUND,He growls and attacks.
640 DATAYou offer him the FOOD and pacify him,He tears you to bits!
650 DATA Somehow you elude him & get away.
660 DATASWORD,STICK,FLY SPRAY,MAGIC POTION,PLATE OF FOOD
670 DATAGUARD ROOM,ENTRANCE HALL,PARLOUR,SCULLERY,MAID'S QUARTERS
680 DATABATHROOM,TOILET,LIBRARY,ANTE ROOM,GREAT GALLERY
690 DATAKING'S CHAMBER,ROYAL BEDROOM,TOWER
700 DATASECRET OUTSIDE PASSAGE,MOAT,TREASURE ROOM
710 DATANORTH,EAST,SOUTH,WEST
720 DATA0,0,0,2,6,1,-1,3,4,2,0,0,0,5,3,0,10,0,0,4,9,7,2,0,0,0,0,6
730 DATA0,0,0,9,0,8,6,10,13,9,5,11,0,10,0,0,0,13,0,0,14,0,10,12
740 DATA15,16,13,0,0,0,0,0,0,0,8,0

```

## Software Review - A.L.F.O.

The letters stand for Alternative Language For Ohios, and indeed, ALFO is a new language for your computer.

ALFO is a Basic program. When loaded and run, it allows you to program in a language rather like assembler, but much easier to learn, a great deal slower, and more forgiving of errors.

It took me about 3 hours to read through the 26 page manual and 7 page instruction set and error codes, load the program, and go through the manual slowly, trying out the examples. There are some 34 commands and statements to learn, and educating yourself in their use is an interesting and refreshing task. Everything worked just as the manual described, and all examples and problems were clearly explained. After a second session, in which I tried a few problems for myself, I felt I was fairly proficient in ALFO programming and editing. A great deal of effort has gone into making the manual easy to read and understand. Suggestions are also offered on ways to change and extend the ALFO codes. For me, one of the most interesting things was to look through the program to see how it had been done.

To summarise:- ALFO is an educational program. As a language it is easy and fun to learn, and is a good preface to assembler programming. It comes with complete documentation and costs \$9.95 including P & P. ALFO can be ordered from Brad Monsborough,

It suits C1 or C4.

# — SUPERBOARD —

## JANUARY PROGRAMMING COMPETITION

I've been mentioning this in SUPERBOARD now for some time. The competition is open to all KAOS members. Rather than send out a form with each newsletter I'll just mention what is required in the way of format, and etc. There are already too many damn forms in this world anyway!

Screen Formats:- Only 24 x 24 standard C1P or 32 x 64 standard C4P.

Program :- Can be a game, educational, utility or anything. The program must not be copyright. Up to 2 programs may be entered.

Medium :- Record 2 copies of each entry at 300 baud, mono, half track in OSI format, on one side of a C60 cassette. Include your name on program and cassette, and mark the cassette "January Competition". Do NOT send the cassette case.

Prizemoney :- \$80 has been allocated. Distribution will depend on the number of entries in each format.

Deadline :- Send the tape plus any documentation needed to use the program to Ohio Superboard User Group, 146 York St, Nundah, 4012. The entry must arrive before January 7, 1984 to be acceptable.

A couple of weeks after the deadline, I will return your tape with all competition entries recorded on it, and a voting form. Once votes are in, the prize winners will be announced. Only those who vote will be eligible to receive prizemoney.

-----

## MORE ON THE TOKEN LOAD/SAVE PROGRAM

The M/C part of the Token Checksum Load program published by John Whitehead in KAOS last month is a truly remarkable piece of code. I'm sure M/C programmers who have disassembled it will have been as impressed as I was by the way it works. Alan Cashin wrote the original program and published it in OSUG newsletter #22, March 1982. Alan is a professional programmer who lives in Armidale, NSW. Recently he set up an ATARI sales outlet, and no doubt has switched his expertise to programs for those machines.

By modifying the program, John Whitehead has improved it considerably to allow for good documentation to be presented on the screen while the program is being loaded.

A month after the program was first published, Alan and his family visited for a couple of days while on a camping holiday to north Queensland. He challenged me to write the program in M/C, and the result, some months later was BASIC 4. It took me three re-writes to jam the code into the available space.

Basic 4 offers up to 60 character spaces that you can use to put program names or directions on the tape. It offers the same high speed format for both BASIC and M/C programs. In addition there is a Verify command, selective loading of any specified program, and you decide whether the program will automatically run when you load it. The program uses the ACIA to do error checking on each byte loaded, and the new Basic 4 is compatible with all systems, including rom based disk operation systems. A special version is available for machines with the ACIA at \$FC00, minus the Verify and Auto-start. Basic 4 has been tested to work at 4800 baud. At 1200 baud, which is attainable with a simple mod, WP-6502 loads in 25 seconds and EXMON in 15 seconds. The old LOAD/SAVE formats are retained, and in addition, a special feature fixes the Warmstart OM error and solves some hangups.

BASIC 4 is available from OSUG for \$12.80 including postage within Aust.

Ed Richardson.



THE BEGINNING MACHINE LANGUAGE PROGRAMMER ....part 17  
*by David Dodds*

We were discussing last month the use of the ACIA to transfer data to and from tape.

The ACIA you may recall contains 4 addressable registers

Receive Data Register	(Read only)
Transmit Data Register	(Write only)
Control Register	(Write only)
Status Register	(Read only)

The Status register is 8 bits wide. Each bit provides some specific information about the ACIA or the data being handled in true/false format.

Of this information we will make use of 3 items

- Bit 0 Receive data register full
- Bit 1 Transmit data register empty
- Bit 6 Parity error

Let us assume that the ACIA has been set to provide  
8 bit data word  
Even parity checking

Given these instructions, the ACIA will arrange to frame the 8 bits of data with a leading start bit and a trailing parity bit and a stop bit ie.

-----  
\* Start Bit \* 8 Character Bits \* Parity Bit \* Stop Bit \*  
-----

Therefore in this format, for each 8 bits of data, the ACIA deals with 11 bits.

Before sending data, it will be necessary to read Bit 1 (the 2nd bit) to see if the ACIA is ready to receive data. This operation will be repeated until the ACIA is ready.

The sequence in assembly language is:

```
1      LDA #2      ;%00000010
2 READY? BIT ACIA
3      BNE READY?
4 SEND  LDA CHAR.
5      STA ACIA+1
```

The BIT instruction you recall is a specialised AND. If Bit 1 of the ACIA Status Register is a 1 then the Z flag in the Status Register of the 6502 will also be 1 and no branch is taken. Otherwise Z will be 0 and the program will loop back to retest the ACIA.

During a read a little more effort is required as it is necessary to check for errors in the received data.

The procedure is:

Test Bit 0 of the ACIA status until a 1 occurs.

Test Bit 6 of the Register.

If Bit 6 is not 0 then an error has occurred.

The following sequence uses the AND instruction to test the ACIA.

```

1 TEST    LDA ACIA
2         TAX      ;save in case data ready
3         AND #1
4         BEQ TEST
5         TXA
5         AND #$40  ;%01000000
6         BNE ERROR
7         LDA ACIA+1

```

Note that this is not the most efficient way of testing the ACIA for data ready. Can you work out how to do the same testing with only 4 lines of code? Hint -use BIT.

---

At last a ROMable Assembler for OSI!

ASM 65 is a MOS Technology format assembler based on the operating principle of the OSI assembler/editor. ASM 65 is comparable in operation to the OSI assembler/editor but provides greatly extended capabilities.

Four assembly options are provided plus optional retention of symbol table between assemblies, meaningful error messages and symbol table generation following assembly.

The resident editor allows full and partial listings of files, block deletion of lines, user defined resequencing, string search and copying of lines within the file. Set assembly offset, load checksum tapes, move data in memory, view tapes without loading, call external machine language routines and manipulate workspace all under keyboard control.

ASM 65 uses no RAM above page 2 so even with only 8K you can do some solid programming.

ASM 65 comes in 2 4K EPROMS with a 24 page operators manual for a miserable \$35 plus postage.

Currently ASM 65 is only available for S2 REV.D machines and uses DABUG 3 as a screen editor. Later versions for machines with 64 character screen formats (including RABBLE 65) will have their own screen editor. The S2 version occupies \$8000-\$9FFF. Other versions at either \$8000-9FFF or \$A000-\$BFFF.

Contact David Dodds

for further

details.

---

SYDNEY OSI GROUP  
by Norman Bate

At our last meeting in September the total attendance was 23 with an average attendance of 16 for the whole day. Number of computers was 7.

We are still endeavouring to find a more central location at a reasonable cost (ie. nothing) but have had no success so far. Suggestions would be welcome.

In the meantime the next meeting will be on Sunday 27th November at 10am in the Lugarno Girl Guides Hall. For those who haven't done so, please bring your computer.

For information contact N. Bate

or N. Bisset

NOTES FROM THE WEST  
*by Wayne Geary*

Once again I have been busy modifying my BASIC and my latest efforts have been the provision of three new commands. From interest shown here at our local group, one of these new commands is much needed by all OSiers.

This new command is an INKEY function which in my system I use to replace the almost worthless NULL command (a POKE will do for those who still need it).

INKEY functions as follows:-

1. Finds or creates the variable
2. Scans keyboard for possible input.
3. Determines if string or variable required and stores character for string or ASCII value if variable.

The routine as I have written it sits at \$0222 and can be operated using the USR function for those who don't wish to alter the BASIC ROMs/EPROMs.

To use the USR function just set the vector in page zero with the usual **POKEs** and tack the variable or string name directly after, ie `X=USR(8)B` will put the ASCII value of the current key pressed into variable 'B' or `X=USR(8)C$` will make `C$` equal to the key pressed. If say the 'A' key was pressed, B would have the value of 65 or `C$` would be "A". If no key was pressed 'B' would equal zero or `C$` would equal "" (null string). Most people should have no difficulty relocating the routine to burn it into EPROM.

The second and third commands are SET and RESET (using &S and &R) which I use with my modified 540 video board and my own character generator chip. In the 64x32 screen mode this gives me a graphics resolution of 128x128 and a fair display without building a Hi-Res board costing \$\$\$'s. My character generator chip uses only 128 characters plus reverse mode to create the full 256 characters required to achieve all combinations needed for the 128x128 resolution. Although I haven't tried it, I believe the same routine with some mods in the maths section could be used with Bernie Will's graphics character set.

I have also rewritten my keyboard polling routine (based on Aardvark version not Rodney Eisfelder's) which simulates a typewriter. Both left and right shift do the same thing and shift-O, shift-P, shift-N etc. are now Repeat-O, P, N etc. for back space, @, ↑, I, L.

Entry at \$FD00 will give normal waiting input whereas entry at \$FD11 will scan and give non-halting input. Returns a null if no key pressed (allowed me to shorten my INKEY routine considerably!)

Also on the rewrite side, I have rewritten WP6502 (insert fanfare) and it now requires less than 2K of memory and will still run in EPROM. All features are still there plus earlier enhancements and only DOS support is gone. With my new keyboard routine as well, ESC is no longer required for upper-lower case shifts and further 20 bytes can be saved. For those with WP6502 and maybe not interested in any of the above I still have a very good hint:- WP6502 will accept all commands in lower case ie. "t" for "Type" or "s" for "S/Ed" will work.

Finally for all those running out of EPROM space, did you know there is

about 1/2K of spare space in DABUG or 3/4K if you don't intend to use disk drives. I'm sure David Anear won't mind having his name deleted in the cause of progress. Actually I rewrote DABUG, including cursor/edit functions, simple monitor and system reset program to suit my system (all subroutines still intact at original addresses) and pruned the code down by about 20% using a few ideas I had. Even some of my own code got the chop and over a weekend I rewrote about 7K and reduced it to 6K!!

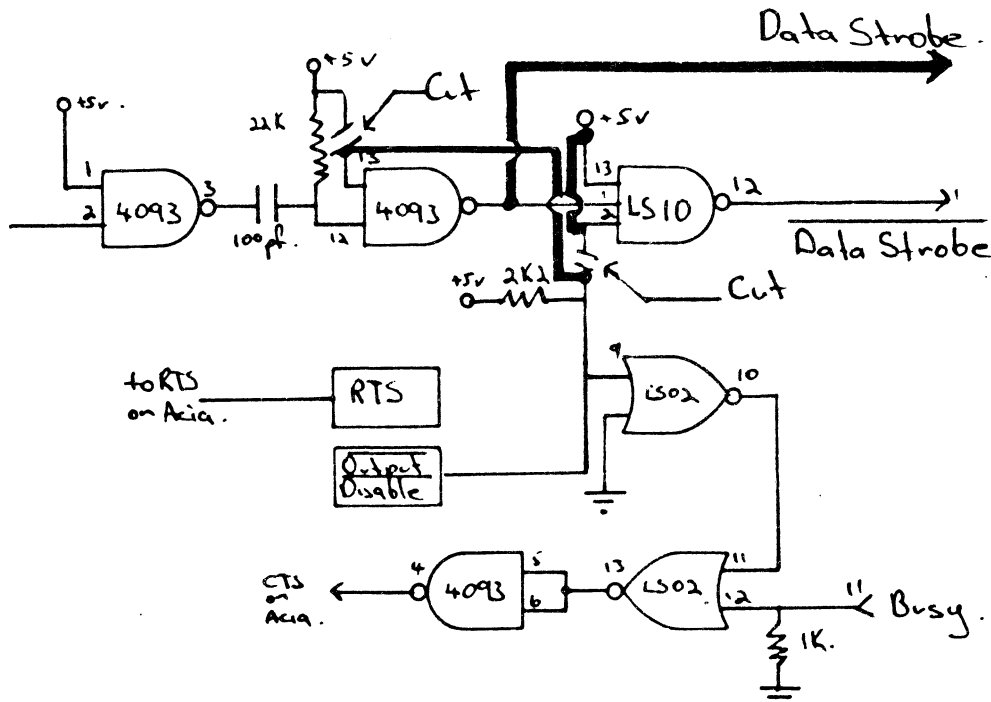
Anyone interested in the keyboard routine, replacing NULL with an INKEY function or the 2K version of WP6502 can write to me at

PS Return postage would be appreciated.

#### FIX FOR GEOFF COHEN'S PRINTER BOARD

*by Jeff Rae*

Many people have contacted me because of problems using the OSI Parallel Interface board with some printers. The difficulty stems from the fact that the data strobe is of the wrong polarity. To fix this make the two cuts and three links as per the diagram. This then gives you a negative or positive strobe.



Check your printer manual for information on the strobe polarity required.

#### THE AUSTRALIAN BEGINNING

Hi there. Well, with apologies to the Commonwealth Bank, The Australian Beginning is going from strength to strength! Not only are membership sales increasing at a fantastic rate, but the system as a whole is growing! We are expecting the delivery of our new baby, a DATA GENERAL MV8000 computer on 18th November. This will not only increase our operating speed greatly, but will also enable us to make long-awaited system operational improvements! Along the same lines we now welcome 2 new members of the T.A.B. staff. These are Jon Ettershank, Development Manager, who will be working alongside David Lutz, and Andrew Lighten, Development Programmer.

Jon Ettershank comes to us from Tandy, where he was a training instructor/customer services representative and was the major influence in designing and setting up the Tandy network on the system. He has a great deal of expertise in many areas, and will be a major addition to our staff.

Andrew Lighten comes to us straight from Ferntree Gully High School, where he has just finished H.S.C. Andrew wrote the TRS-80 front-end and the chat program on T.A.B. as well as many system improvements and will greatly enhance the programming side of our operation.

Another addition to our service will be the long-awaited connection to Telecom Australia's Austpac Network. Access to the system via Austpac will enable all members within Australia to be on an equal footing with each other and the high S.T.D. charges will be a thing of the past for the more remote users! For more information on this topic type AUSTPAC at any sel( ): prompt when logged onto the system.

There are many changes being made to the system at the moment and this will be an on-going function of T.A.B. We are responsive to you, the user, and suggestions for improving our service are always welcome via mail or through suggestion box on the system.

We would also like to take this opportunity to wish all readers of the KAOS newsletter a happy Christmas and a prosperous New Year.

Noel Fenton  
Chief Executive Officer

David Lutz  
Customer Services Manager

---

---

NMhz FIX  
*by Jeff Rae*

Have you ever wondered why you have to hold down the Repeat key to get the OS65D disk to boot when using the Tasan video board and a Superboard with a C4-P keyboard.

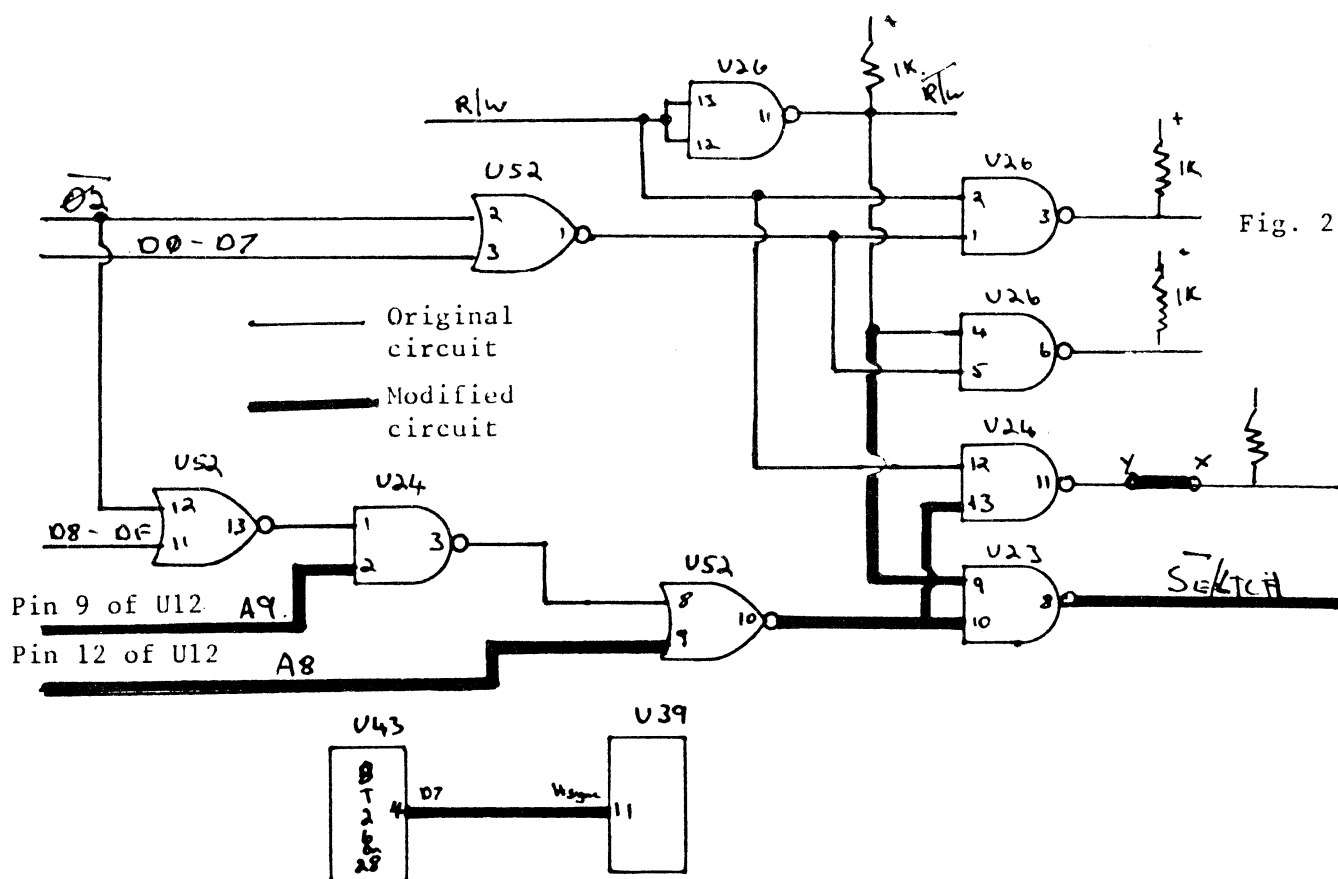
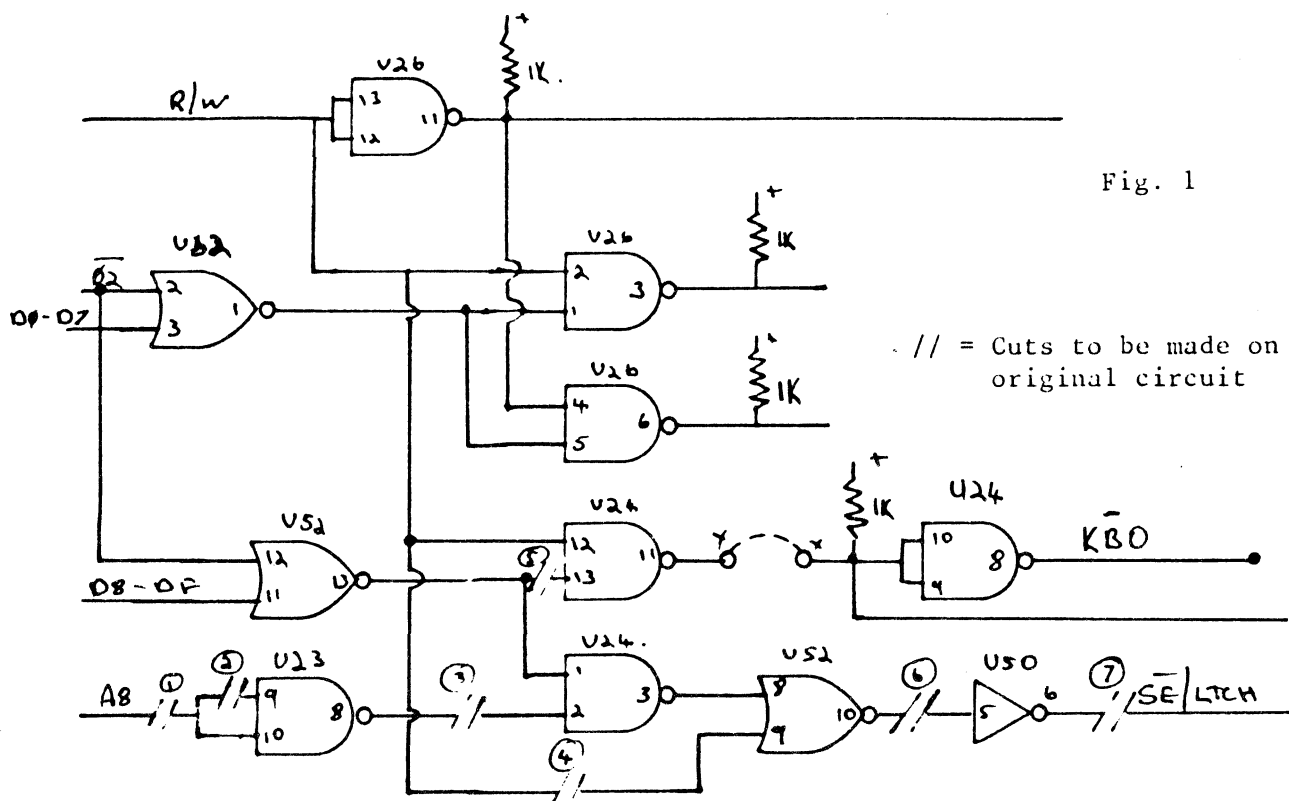
The reason is that the OS65D has a routine that is used to determine what speed the CPU is running at, 1 Mhz or 2 Mhz.

To do this, the software looks for a 60 hz signal which is connected to D7 at location DE00. It was found that due to Superboard's partial address decoding the keyboard was not only addressed at DF00 but also at DE00. If you check the schematics you will find that pressing Repeat puts a high on D7.

The main disadvantage of this is when you run your system at 2Mhz. The DOS routine will decode the Repeat key as a reference signal but it will set itself up as though it is running at 1Mhz and this will cause trouble when you try to write information to disk.

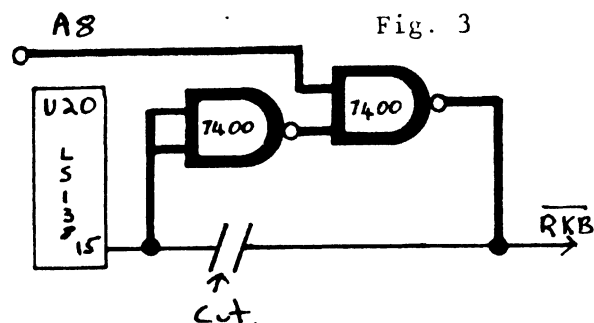
One for remedy this is to permanently change DOS so that it runs correctly at 2Mhz. The problem is that you then have to change all your disks .

The other method is to change the Tasan video board so that it generates the needed NMhz signal. Another bonus of the modification is that also fixes the double addressing of the video latch so that it only appears at DE00-DEFF.



While the above circuit generates the NMhz signal on the Tasan video board you still have to press the Repeat key to boot a disk as the addressing for the keyboard on the Superboard needs modification so that it only decodes DF00-DFFF.

To do this bend all pins except pins 7 and 14 of a 7400 so that they are parallel to the Ic. Then solder it piggy back style on another 74xx series Ic with the same power pins, consult TTL handbook. Cut the trace from pin 15 of U20 and run the wires back to the 7400 chip as per the following diagram, consult TTL handbook for pin numbers.



You will find that there is no need to press the Repeat key when booting your Disks.

### IMPROVED 3.3 SCREEN DUMP *by Leo Jankowski*

In a recent issue of KAOS someone complained about the unusability of the screen to printer dump in OS65D 3.3. I wrote the following screen to printer dump program to run with BASIC in ROM. I tried it in OS65D 3.3, and it worked fine. Maybe the printing could be speeded up by re-writing the code, leaving out the zero-page indexed addressing bits.

9D2B 40	9D49 C8	INY
9D2C 1E	9D4A C000	CPY #\$00
9D2D 00	9D4C D002	BNE \$9D50
9D2E D0	9D4E E6F4	INC \$F4
9D2F 00	9D50 E4F1	CPX \$F1
9D30 A200 LDX #\$00	9D52 D0EF	BNE \$9D43
9D32 BD2B9D LDA \$9D2B,X	9D54 A90A	LDA #\$0A
9D35 95F1 STA \$F1,X	9D56 20B1FC JSR \$FCB1	
9D37 E8 INX	9D59 A90D	LDA #\$0D
9D38 E006 CPX #\$06	9D5B 20B1FC JSR \$FCB1	
9D3A D0F6 BNE \$9D32	9D5E E6F5	INC \$F5
9D3C EE0502 INC \$0205	9D60 A5F2	LDA \$F2
9D3F A000 LDY #\$00	9D62 C5F5	CMP \$F5
9D41 A200 LDX #\$00	9D64 F002	BEQ \$9D68
9D43 B1F3 LDA (F3),Y	9D66 D0D9	BNE \$9D41
9D45 20B1FC JSR \$FCB1	9D68 CE0502	DEC \$0205
9D48 E8 INX	9D6B 60	RTS

Registered by Australia Post  
Publication No. VB64212

If undeliverable return to  
KAOS, 10 Forbes St  
Essendon, Victoria 3040

KAOSKAOS  
K Postage K  
A Paid A  
O Essendon O  
S 3040 S  
KAOSKAOS